Grid Protocol Architecture Research Group
GGF-5 Jul 22, 2002 2:30 AM PDT

Chair: Bill Johnston

Proposed reorganization of GPA-RG in recognition of the new Open Grid Services Architecture Group. Need use cases to help drive prototyping.

OGSA BOF is establishing a road map for the working groups needed to implement Open Grid Services Architecture.

Will define services for key grid functions including security, domain management.

Proposed new working group:

Original charter was too ambitious.
Now want to define the minimum services that make up a grid, such that they are needed by every grid application, and such that they cannot be created from lower level services.

Will discuss the draft at the meeting
Complete the draft by GGF-6, submit for full discussion at GGF-7, and submit for final public comment at GGF-8.

Document review:

Focus on defining functions without specifying interface. Can functions be defined such that any grid will work (C environment, OGSA environment, etc.)

Need operational support for the key functions, install and maintain servers for production system.

URL: http://www-itg.lbl.gov/~wej/files/GCS.v3.doc

Basic services:

1) Grid information service – discovery mechanism
2) Resource scheduling – binding components before execution
3) Uniform computing access – initiate processes on computers
4) Uniform Data Access – assumed transport is the only mechanism
5) Asynchronous Message Service – event mechanism
6) Remote authentication, certificate management
7) Grid communication – communication abstraction
8) System management – GSI enabled SSHD

Other proposed services

Transaction management service – candidate for inclusion in Asynchronous message or in Grid communication

Asynchronous, synchronous, and multi-phase commit
Mechanism to survive TCP/IP communication breakdowns

Want interoperability between transaction management systems

Is there a difference between distributed computing and grid computing? Is XA semantics the right semantics for distributed transactions? Problem is that XA does not scale beyond 3 platforms.

OGSA attempts to use services to describe interactions. Can use state information, which differentiates from WSDL. Are working on WSTX transactions. What is occurring with respect to WSTX for web services, and can this be used in a grid environment where the services have state. OGSA does not address transactional integrity.

Database access should include spatial, time series, which do not rely on standard SQL.

What is the difference between application support, operational support for XA, and application communicating over a grid. Is system support needed. XA is a protocol that uses two phase commit. Requires that each application use an error recovery, but does not scale well. When introduce distributed state, does this change the protocol?

Grid has focused on distributed batch processing, rather than transactional semantics. As grids become commercially viable to support distributed computing, need transactional support. Need coordinated recovery such as checkpointing and restart as alternate to transactional semantics.

Need input from the GGF checkpoint group as well as WSTX to decide if transaction management is one of the minimal services.

A service that needs persistent state requires operational support, and should be one of the minimal services. Will transaction coordination require persistent state?

What kind of persistent state is required? A general information service may suffice. Do we need more than one registery? Scalability is a separate issue.

Minimal services need persistent state, but cannot be built on other services, with the exception of security. A similar global service is the logical name space for data within the grid.

Policy for resource provisioning and agreement on language for policy, schema and schema transformation, are broader than authorization. WSLA is a language for defining service level agreements. May be able to convert from this language to resource allocation.

Uniform data access is composed out of a logical name space and a storage repository abstraction. Think of GridFTP as opaque type data access. Access to time series is very specific data model, that requires different characterization. Can talk to databases to retrieve XML file for

> all data types. Want mechanism to retrieve XML defined object. Need standard information repository abstraction for the extraction of the XML object. Need protocol for how the data will be returned.
>
> Do we need to differentiate between naming in general, and name abstraction definition for XML objects? Need to specify data type specification. Below uniform data access, have a generic XML format for opaque types.
>
> OGSA is introducing common resource models used in grid space. Examples are MDS discovery service, and the resources they operate on. Each grid approach has a different resource model, and a specific protocol for talking to the resources. Want common models that can publish to a registery, and can support different binding protocols. Is there an overlap for a data abstraction for how can reference and extract.
>
> Reliable transaction and reliable recall are different. Are storage area network management of volatile state management important? Data storage semantics within SANs requires coordination of data state. Need to externalize behavior for transactional latency of storage across caches. Need storage abstraction model for reliable data store. Is the degree of replication the important criteria? How handle latencies when store data.
>
> Need abstraction for the underlying computing mechanism. Assumes that where the script is running, can get at tertiary storage, access data, and put onto local file system before execution starts. What are the variations of this problem? May assume that users have home directories on AFS. Want specific examples of the assumed name space, location of data, and access mechanisms to data before computation takes place.

Use case model for Unicore

Differentiate between Resource Requests and Resource Providers. There can be a one-to-many mapping from request to provider. There can be many-to-many mappings.

To build a grid economy, map resource request to resource providers, constrained by quality of service.

Unicore promotes seamless access. Do not know what resources will use. Uses a modeling paradigm, to create an abstraction of workflow process, modeled as Acyclic graph. Uses an Incarnation Database to track mapping to resources.

Compose applications from plug-ins. Abstract job object are sent to gateway for security checking. All subsequent operations are done through sockets. A target system interface is produced, which is the script that runs the job.

Have job abstraction, incarnation, file staging and transfer support.
Use X509 certificates, multiple CAs
Generic clients modified by plug ins
Written in Perl and Java

Comparison to Globus:
Same model for discovery and request in Unicore
Workflow environment instead of APIs and toolkit
End to end security model, not built on transitive trust.
Incarnation of abstractions at server

Can put Unicore as a workflow portal on Globus, using GSI. Are mapping DLAP to Unicore resources.

Use Globus mechanisms to talk to machines, at level of TSI. Were going to create Incarnation database dynamically by querying Globus IDB. This would allow ephemeral changes to be tracked.

Critical functions:
Authentication
Compose workflow
Incarnation of abstract workflow
Needed ability to discover resources. Created a uniform resource broker to interrogate a site to find if can run a job. Extends Unicore functionality.

Translation process requires an ontology to manage multiple workflow languages. Ontology used for actual mapping to resource processing.

Unicore builds upon a generic job model. Has an abstraction of the architecture for a batch job model. Can extend Java classes for alternate architectures. Could do some of the architecture specification at the TSI layer. Need a characterization for each supercomputer center architecture. The incarnation database stores some of the translation information.

Is the workflow model sufficiently sophisticated? Is there a meta-flow workflow model, or a mechanism to translate between workflow models? Expect clues to the meta-flow to be found from the ontology for translating between workflows.